

Noisebridge Neural Networks

Christoph Maier

Neuron

■ General neuron: nonlinear function of scalar product

```
neuron = Function[{input, weights, sigmoid}, sigmoid[weights.input]]  
  
Function[{input, weights, sigmoid}, sigmoid[weights.input]]
```

■ Unit step neuron

```
stepneuron =  
  Function[{input, weights, threshold}, neuron[input, weights, UnitStep[# - threshold] &]]  
  
Function[{input, weights, threshold}, neuron[input, weights, UnitStep[#1 - threshold] &]]
```

Learning

```
educate = Function[{input, target, nerve, weights, tweak, rate},  
  {weights + rate input (target - nerve[input, weights]), tweak[rage]}]  
  
Function[{input, target, nerve, weights, tweak, rate},  
  {weights + rate input (target - nerve[input, weights]), tweak[rage]}]
```

Invariants under learning iteration: type of neuron, Function how to tweak learning rate

Iterated: weight vector, learning rate

Variant inputs: Learning vector, learning target

```
educatiter = Function[{invariant, iterated, variant},  
  Block[  
    {nerve = invariant[[1]], ratetweak = invariant[[2]],  
    input = variant[[1]], target = variant[[2]],  
    weights = iterated[[1]], rate = iterated[[2]],  
    output},  
    Print[output = educate[input, target, nerve, weights, ratetweak, rate]];  
    output  
  ]]  
  
Function[{invariant, iterated, variant},  
  Block[{nerve = invariant[[1]], ratetweak = invariant[[2]], input = variant[[1]],  
    target = variant[[2]], weights = iterated[[1]], rate = iterated[[2]], output},  
    Print[output = educate[input, target, nerve, weights, ratetweak, rate]]; output]]
```

Training set

```
set = {{{1, 0, 0}, 1}, {{1, 0, 1}, 1}, {{1, 1, 0}, 1}, {{1, 1, 1}, 0}}  
  
{{{1, 0, 0}, 1}, {{1, 0, 1}, 1}, {{1, 1, 0}, 1}, {{1, 1, 1}, 0}}
```

Learning

■ Single learning step

```
educate[{1, 0, 0}, 1, stepneuron[#1, #2,  $\frac{1}{2}$ ] &, {0, 0, 0}, .95 # &, .8]
{{0.8, 0, 0}, 0.76}
```

■ Single learning step with iterable parameters

```
educatiter[{stepneuron[#1, #2,  $\frac{1}{2}$ ] &, .95 # &}, {{0, 0, 0}, .8}, {{1, 0, 0}, 1}]
{{0.8, 0, 0}, 0.76}
{{0.8, 0, 0}, 0.76}
```

■ Run once over the training set

```
Fold[educatiter[{stepneuron[#1, #2,  $\frac{1}{2}$ ] &, .99 # &}, #1, #2] &, {{0, 0, 0}, .8}, set]
{{0.8, 0, 0}, 0.792}
{{0.8, 0, 0}, 0.78408}
{{0.8, 0, 0}, 0.776239}
{{0.0237608, -0.776239, -0.776239}, 0.768477}
{{0.0237608, -0.776239, -0.776239}, 0.768477}
```

■ Run 10 times over the training set

```
Nest[Fold[educatiter[{stepneuron[#1, #2,  $\frac{1}{2}$ ] &, .99 # &}, #1, #2] &, #, set] &,
{{0, 0, 0}, .8}, 10]
{{0.8, 0, 0}, 0.792}
{{0.8, 0, 0}, 0.78408}
{{0.8, 0, 0}, 0.776239}
{{0.0237608, -0.776239, -0.776239}, 0.768477}
{{0.792238, -0.776239, -0.776239}, 0.760792}
{{1.55303, -0.776239, -0.0154472}, 0.753184}
{{1.55303, -0.776239, -0.0154472}, 0.745652}
{{0.807377, -1.52189, -0.761099}, 0.738196}
{{0.807377, -1.52189, -0.761099}, 0.730814}
```

```

{{1.53819, -1.52189, -0.0302856}, 0.723506}
{{2.2617, -0.798386, -0.0302856}, 0.716271}
{{1.54543, -1.51466, -0.746556}, 0.709108}
{{1.54543, -1.51466, -0.746556}, 0.702017}
{{1.54543, -1.51466, -0.746556}, 0.694997}
{{2.24042, -0.81966, -0.746556}, 0.688047}
{{1.55238, -1.50771, -1.4346}, 0.681166}
{{1.55238, -1.50771, -1.4346}, 0.674355}
{{2.22673, -1.50771, -0.760248}, 0.667611}
{{2.22673, -1.50771, -0.760248}, 0.660935}
{{2.22673, -1.50771, -0.760248}, 0.654326}
{{2.22673, -1.50771, -0.760248}, 0.647782}
{{2.22673, -1.50771, -0.760248}, 0.641304}
{{2.22673, -1.50771, -0.760248}, 0.634891}
{{2.22673, -1.50771, -0.760248}, 0.628543}
{{2.22673, -1.50771, -0.760248}, 0.622257}
{{2.22673, -1.50771, -0.760248}, 0.616035}
{{2.22673, -1.50771, -0.760248}, 0.609874}
{{2.22673, -1.50771, -0.760248}, 0.603775}
{{2.22673, -1.50771, -0.760248}, 0.597738}
{{2.22673, -1.50771, -0.760248}, 0.59176}
{{2.22673, -1.50771, -0.760248}, 0.585843}
{{2.22673, -1.50771, -0.760248}, 0.579984}
{{2.22673, -1.50771, -0.760248}, 0.574184}
{{2.22673, -1.50771, -0.760248}, 0.568443}
{{2.22673, -1.50771, -0.760248}, 0.562758}
{{2.22673, -1.50771, -0.760248}, 0.557131}
{{2.22673, -1.50771, -0.760248}, 0.551559}
{{2.22673, -1.50771, -0.760248}, 0.546044}
{{2.22673, -1.50771, -0.760248}, 0.540583}
{{2.22673, -1.50771, -0.760248}, 0.535177}
{{2.22673, -1.50771, -0.760248}, 0.535177}

```

■ Sanity check

```
stepneuron[#1, First[%],  $\frac{1}{2}$ ] & /@ (First /@ set)

{1, 1, 1, 0}
```

■ Terminate automatically

```
NestWhile[Fold[educatiter[{stepneuron[#1, #2,  $\frac{1}{2}$ ] &, .99 # &}, #1, #2] &, #, set] &,
  {{0, 0, 0}, .8}, First[#1] ≠ First[#2] &, 2]

{{0.8, 0, 0}, 0.792}
{{0.8, 0, 0}, 0.78408}
{{0.8, 0, 0}, 0.776239}
{{0.0237608, -0.776239, -0.776239}, 0.768477}
{{0.792238, -0.776239, -0.776239}, 0.760792}
{{1.55303, -0.776239, -0.0154472}, 0.753184}
{{1.55303, -0.776239, -0.0154472}, 0.745652}
{{0.807377, -1.52189, -0.761099}, 0.738196}
{{0.807377, -1.52189, -0.761099}, 0.730814}
{{1.53819, -1.52189, -0.0302856}, 0.723506}
{{2.2617, -0.798386, -0.0302856}, 0.716271}
{{1.54543, -1.51466, -0.746556}, 0.709108}
{{1.54543, -1.51466, -0.746556}, 0.702017}
{{1.54543, -1.51466, -0.746556}, 0.694997}
{{2.24042, -0.81966, -0.746556}, 0.688047}
{{1.55238, -1.50771, -1.4346}, 0.681166}
{{1.55238, -1.50771, -1.4346}, 0.674355}
{{2.22673, -1.50771, -0.760248}, 0.667611}
{{2.22673, -1.50771, -0.760248}, 0.660935}
{{2.22673, -1.50771, -0.760248}, 0.654326}
{{2.22673, -1.50771, -0.760248}, 0.647782}
{{2.22673, -1.50771, -0.760248}, 0.641304}
{{2.22673, -1.50771, -0.760248}, 0.634891}
{{2.22673, -1.50771, -0.760248}, 0.628543}
{{2.22673, -1.50771, -0.760248}, 0.628543}
```

Sanity check

```
stepneuron[#1, First[%],  $\frac{1}{2}$ ] & /@ (First /@ set)

{1, 1, 1, 0}

stepneuron[#1, {.5, -.1, -.1},  $\frac{1}{2}$ ] & /@ (First /@ set)

{1, 0, 0, 0}
```